



HP Caliper

A Powerful Itanium® Performance Tool



Curt Wohlgemuth, Steve Williams
HP Caliper Development Team
Hewlett-Packard



Agenda

- Quick overview of Caliper
- Some examples
- New stuff

Quick Overview

- Performance measurement & analysis tool
- Itanium native code
- Linux and HP-UX
- Lots of different measurements using the Itanium PMU
- Per-task (follow processes) and per-CPU (system-wide)
- Measures **any** application

Some example measurements

Basic command line syntax:

```
caliper <measurement> [options] application ....
```

-- or --

```
caliper <measurement> [options] --attach=<pids>
```

-- or --

```
caliper <measurement> [options] --scope=system
```

e.g.:

```
caliper scgprof -p sweep3d script_file
```

```
caliper fprof --scope=system --duration=60
```

Measurement types

Totals: **ecount, cpu**

Profiles: **alat_miss, branch_prediction,**
dcache_miss, dtlb_miss, fprof,
icache_miss, itlb_miss

Call graph: **scgprof**

Traces: **pmu_trace**

What?
(totaled PMU)
<< 1% intrusion

Where?
(sampled PMU)
~ 1% - 3% intrusion

HowTo - One interesting scenario

- Poor man's call stack near an event
 - Use pmu_trace, sample on DATA_EAR_EVENTS, collect BTB
 - No histogramming, just a raw trace:

PMU Trace Buffer (1 of 2), Kernel Thread ID 26462, Samples 1 - 1

Sample Number	Data Cache Miss			BTB Samples			
	Address:Slot (module:function)	Data Address	Latency	Address:slot (module:function)	Type	FE Misp	BE Misp
1	0x40000000004767e0:0 (caliper::sqlite3VdbeExec)	0x60000000002b6f90	5	0x40000000004354e0:2 (caliper::initPage)	Src	No	No
				0x40000000004355d0:0 (caliper::getAndInitPage)	Tgt	No	No
				0x40000000004355e0:2 (caliper::getAndInitPage)	Src	No	No
				0x40000000004360f0:0 (caliper::sqlite3BtreeCursor)	Tgt	No	No
				0x4000000000436220:2 (caliper::sqlite3BtreeCursor)	Src	No	No
				0x4000000000476750:0 (caliper::sqlite3VdbeExec)	Tgt	No	No
				0x40000000004767e0:2 (caliper::sqlite3VdbeExec)	Src	No	No
				0x4000000000433dd0:0 (caliper::sqlite3BtreeFlags)	Tgt	No	No

HowTo - Tuning Strategy

Step 1 - Identify problem

Algorithmic issues

- scgprof
- fprof

CPU bottleneck

- ecount
- cpu

Step 2 - Pin-point location of problem

- fprof / scgprof
- dcache_miss / dtlb_miss
- icache_miss / itlb_miss
- <customized>

Step 3 - Fix, goto step 1

New feature: cpu metrics

- Measure and report groups of CPU metrics
- Implemented using multiplexed metrics sets
 - requires Perfmon2.2 features
- Great as a “first cut” measurement
- Example metrics sets
 - CPI
 - stalls
 - cache miss data
 - overview: most useful metric sets in one run

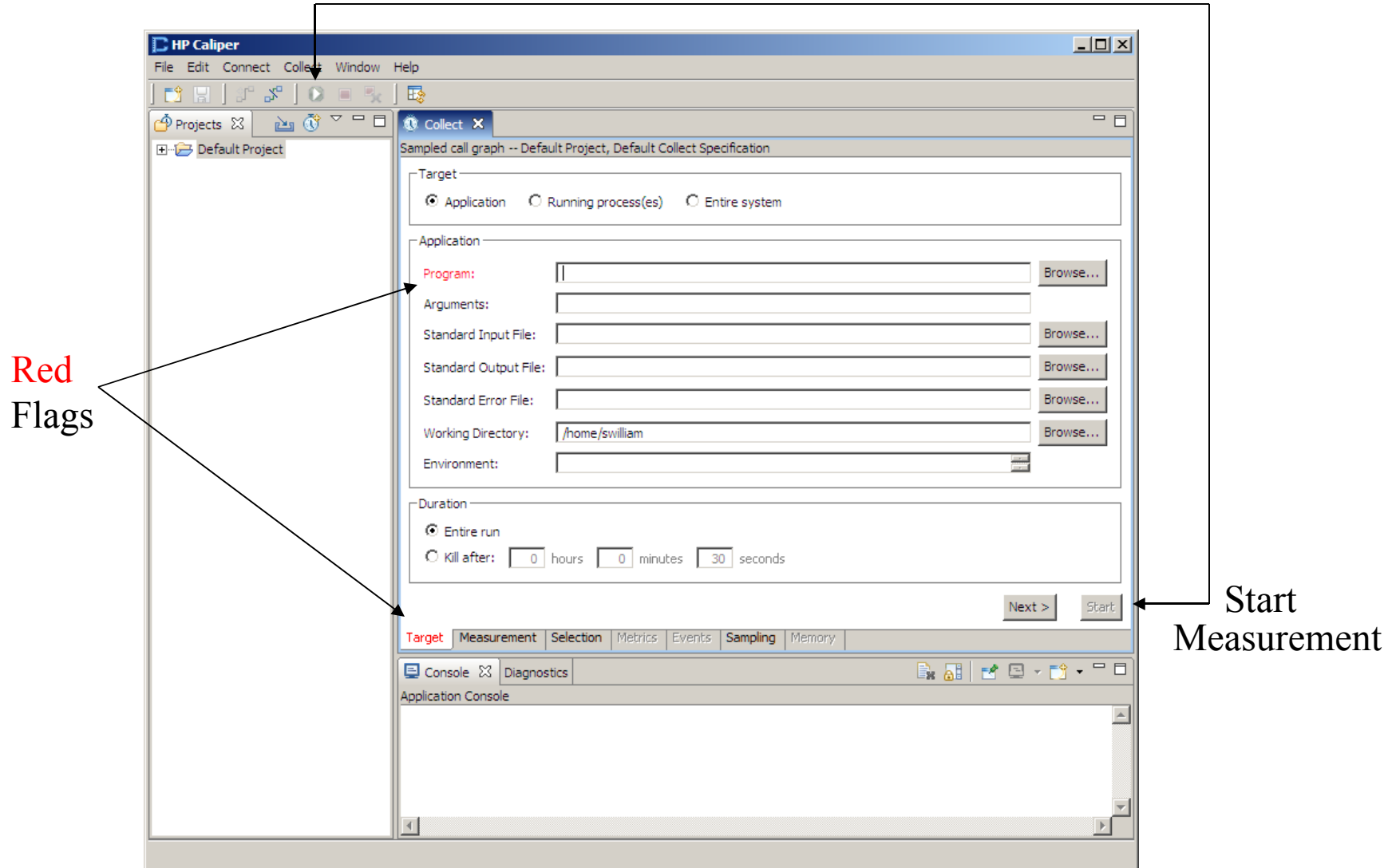
New feature: system-wide improvements

- Can use any measurement
- Measure all CPUs, for counts and/or samples
- Samples can be attributed to
 - individual processes
 - individual load modules within processes
 - to no processes at all
- Attribution to modules gives details down to assembly code

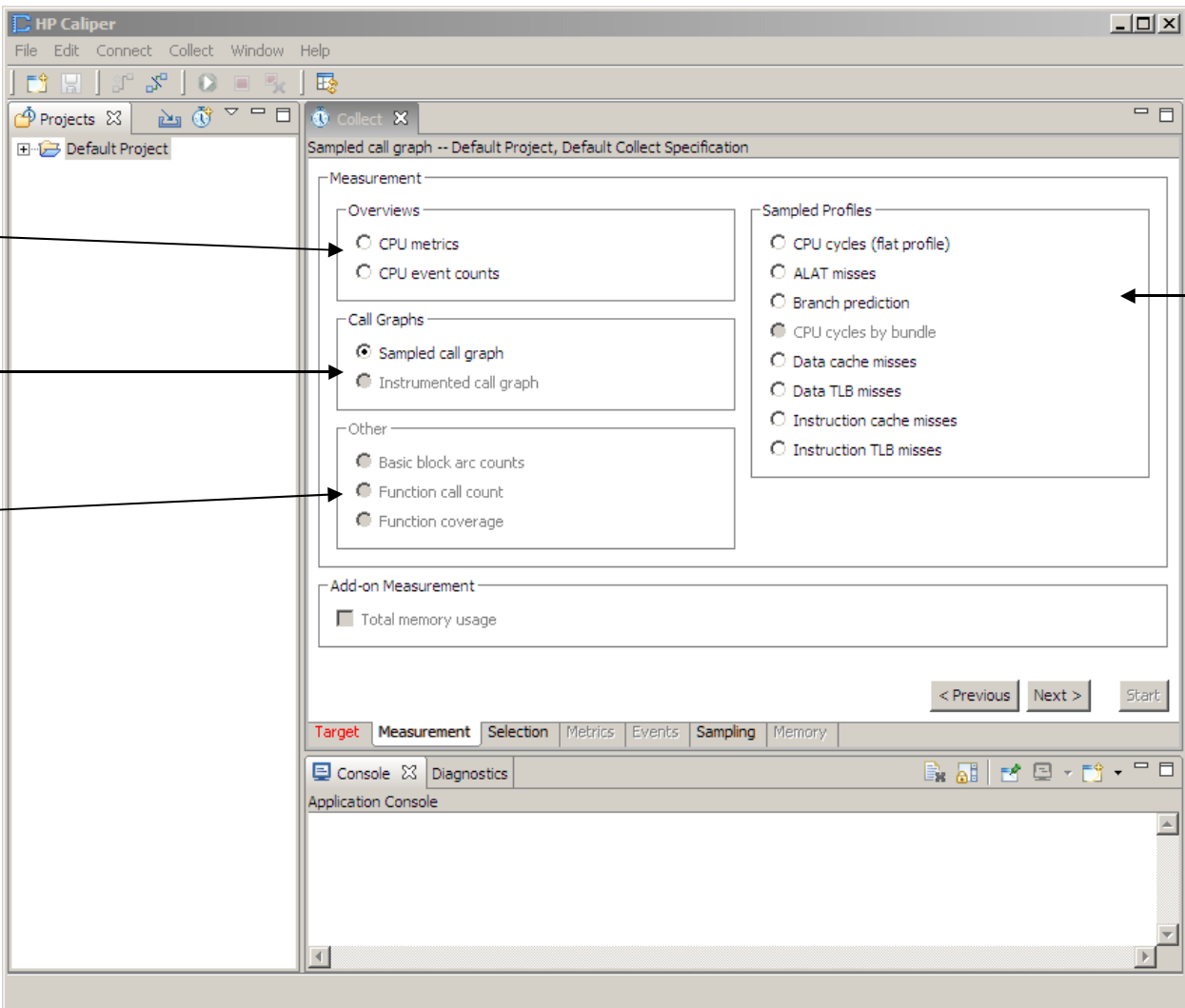
New feature: GUI

- Eclipse RCP-based client
- Connect to Caliper server, to
 - perform measurement collections
 - browse Caliper databases
 - see measurement data, with easy drill down

Initial Screen: Follow Red Flags



Choosing Measurements



The screenshot shows the HP Caliper 'Collect' window. The 'Measurements' section on the left contains three sub-sections: 'Overviews' with 'CPU metrics' and 'CPU event counts'; 'Call Graphs' with 'Sampled call graph' (selected) and 'Instrumented call graph'; and 'Other' with 'Basic block arc counts', 'Function call count', and 'Function coverage'. The 'Add-on Measurement' section at the bottom has 'Total memory usage'. The 'Sampled Profiles' section on the right lists: 'CPU cycles (flat profile)', 'ALAT misses', 'Branch prediction', 'CPU cycles by bundle' (selected), 'Data cache misses', 'Data TLB misses', 'Instruction cache misses', and 'Instruction TLB misses'. Navigation buttons '< Previous', 'Next >', and 'Start' are at the bottom right. A tab bar at the bottom includes 'Target', 'Measurement' (active), 'Selection', 'Metrics', 'Events', 'Sampling', and 'Memory'. The 'Application Console' is at the very bottom.

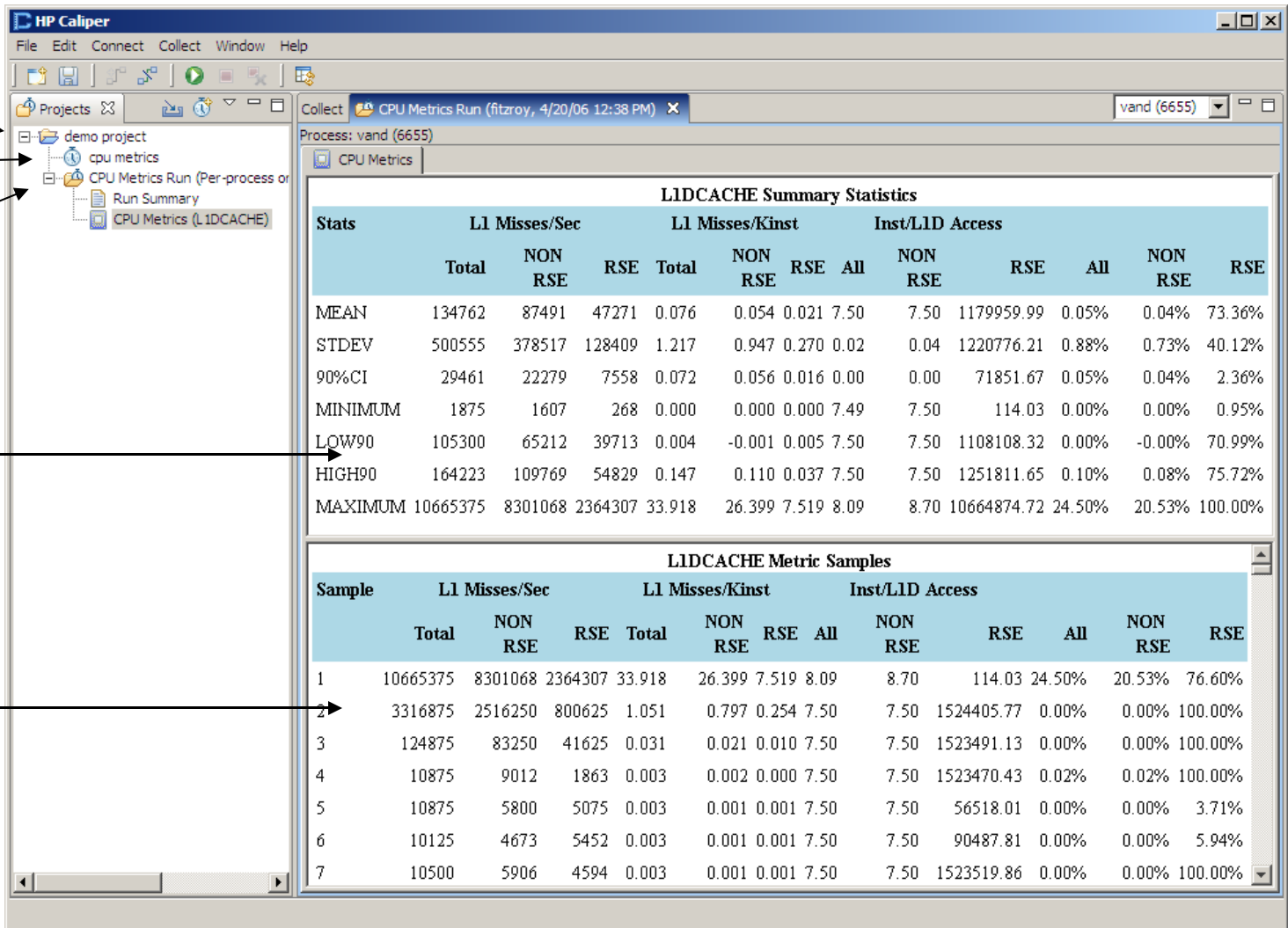
CPU Behavior

Call Graph

Not Available

Profiles

CPU Metrics (L1 Data Cache)



Project

Collection Setup

Collected Data

Summary Statistics

Samples

L1DCACHE Summary Statistics

Stats	L1 Misses/Sec			L1 Misses/Kinst			Inst/L1D Access					
	Total	NON RSE	RSE	Total	NON RSE	RSE	All	NON RSE	RSE	All	NON RSE	RSE
MEAN	134762	87491	47271	0.076	0.054	0.021	7.50	7.50	1179959.99	0.05%	0.04%	73.36%
STDEV	500555	378517	128409	1.217	0.947	0.270	0.02	0.04	1220776.21	0.88%	0.73%	40.12%
90%CI	29461	22279	7558	0.072	0.056	0.016	0.00	0.00	71851.67	0.05%	0.04%	2.36%
MINIMUM	1875	1607	268	0.000	0.000	0.000	7.49	7.50	114.03	0.00%	0.00%	0.95%
LOW90	105300	65212	39713	0.004	-0.001	0.005	7.50	7.50	1108108.32	0.00%	-0.00%	70.99%
HIGH90	164223	109769	54829	0.147	0.110	0.037	7.50	7.50	1251811.65	0.10%	0.08%	75.72%
MAXIMUM	10665375	8301068	2364307	33.918	26.399	7.519	8.09	8.70	10664874.72	24.50%	20.53%	100.00%

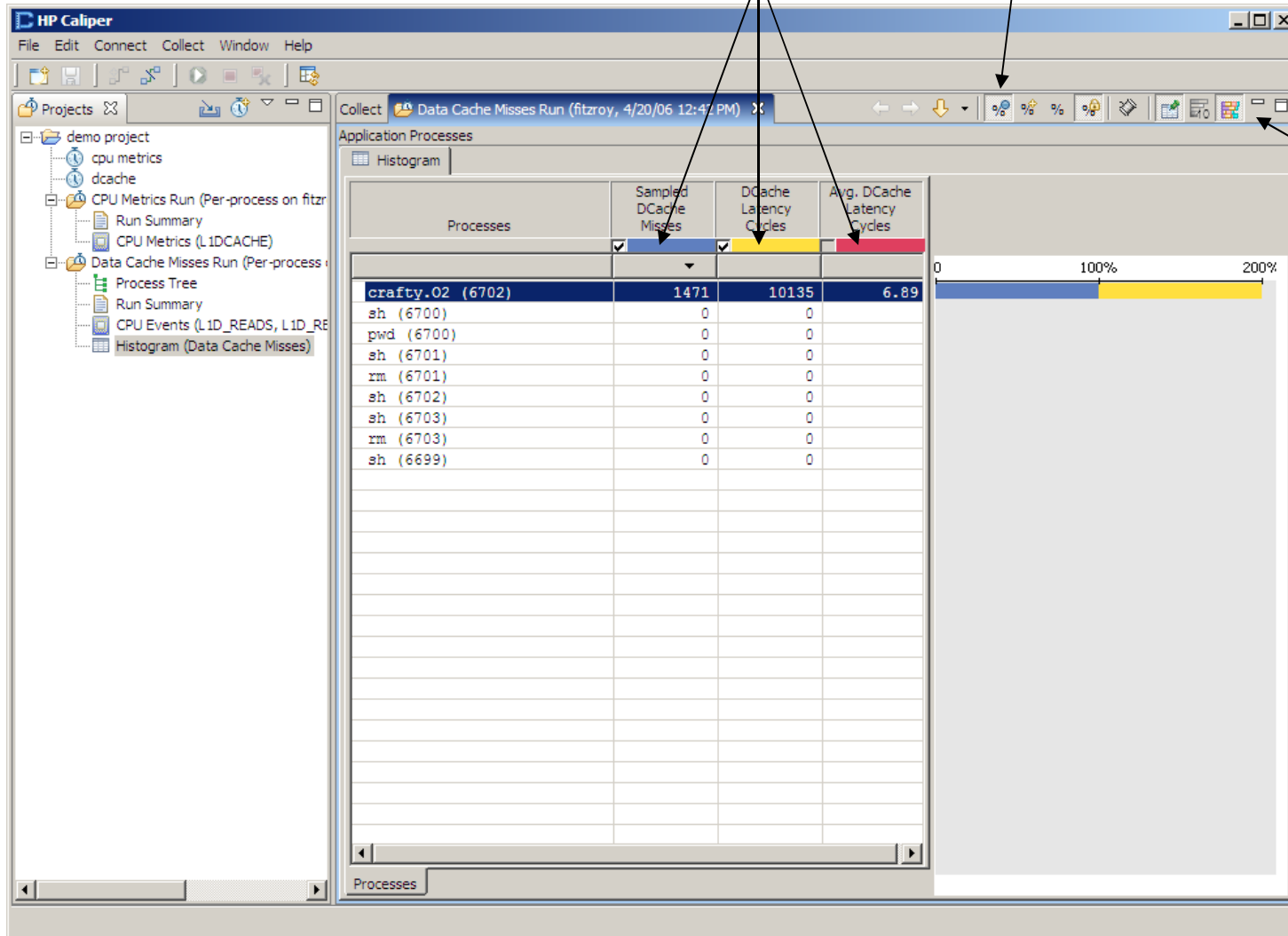
L1DCACHE Metric Samples

Sample	L1 Misses/Sec			L1 Misses/Kinst			Inst/L1D Access					
	Total	NON RSE	RSE	Total	NON RSE	RSE	All	NON RSE	RSE	All	NON RSE	RSE
1	10665375	8301068	2364307	33.918	26.399	7.519	8.09	8.70	114.03	24.50%	20.53%	76.60%
2	3316875	2516250	800625	1.051	0.797	0.254	7.50	7.50	1524405.77	0.00%	0.00%	100.00%
3	124875	83250	41625	0.031	0.021	0.010	7.50	7.50	1523491.13	0.00%	0.00%	100.00%
4	10875	9012	1863	0.003	0.002	0.000	7.50	7.50	1523470.43	0.02%	0.02%	100.00%
5	10875	5800	5075	0.003	0.001	0.001	7.50	7.50	56518.01	0.00%	0.00%	3.71%
6	10125	4673	5452	0.003	0.001	0.001	7.50	7.50	90487.81	0.00%	0.00%	5.94%
7	10500	5906	4594	0.003	0.001	0.001	7.50	7.50	1523519.86	0.00%	0.00%	100.00%

Histogram Viewer (Top Level)

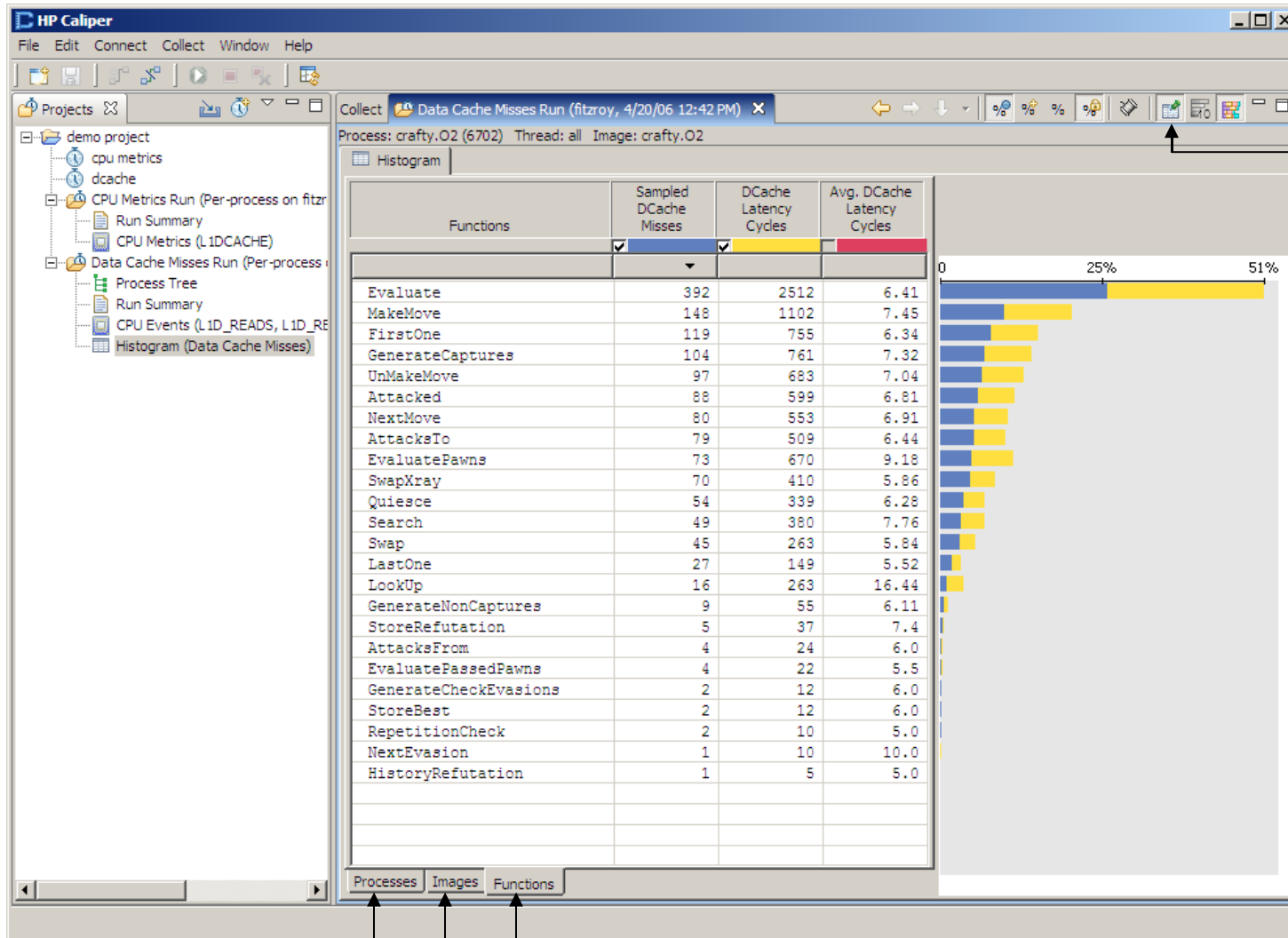
Bar Chart Legend/Chooser

Show Percent of Grand Totals



Use Stacking Bar Chart

Function Histogram (Data Cache)

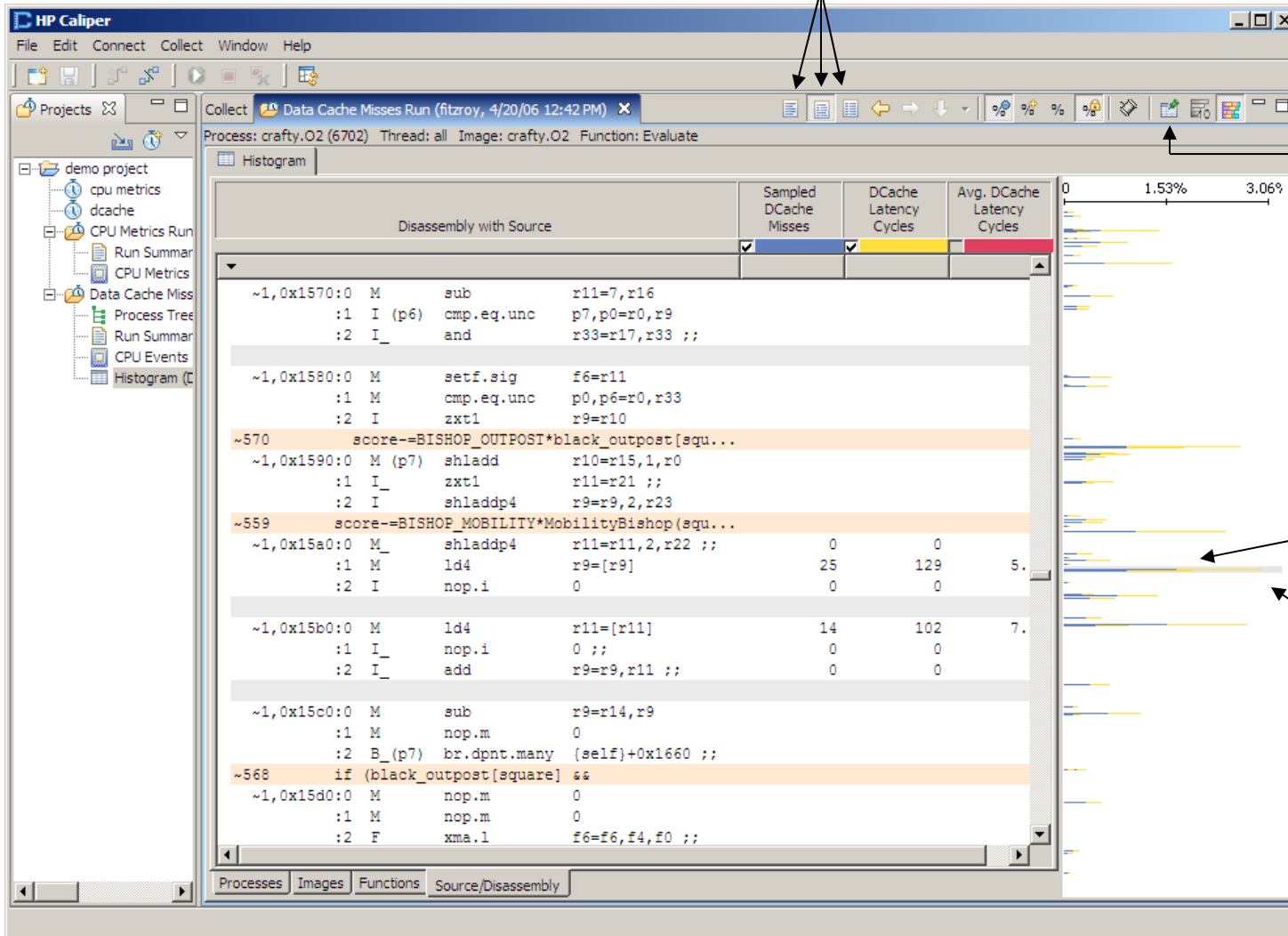


“Pin” Graph to Table

Drill-down History

Disassembly with Overview Graph

Modes: Source, Disassembly with Source, Disassembly



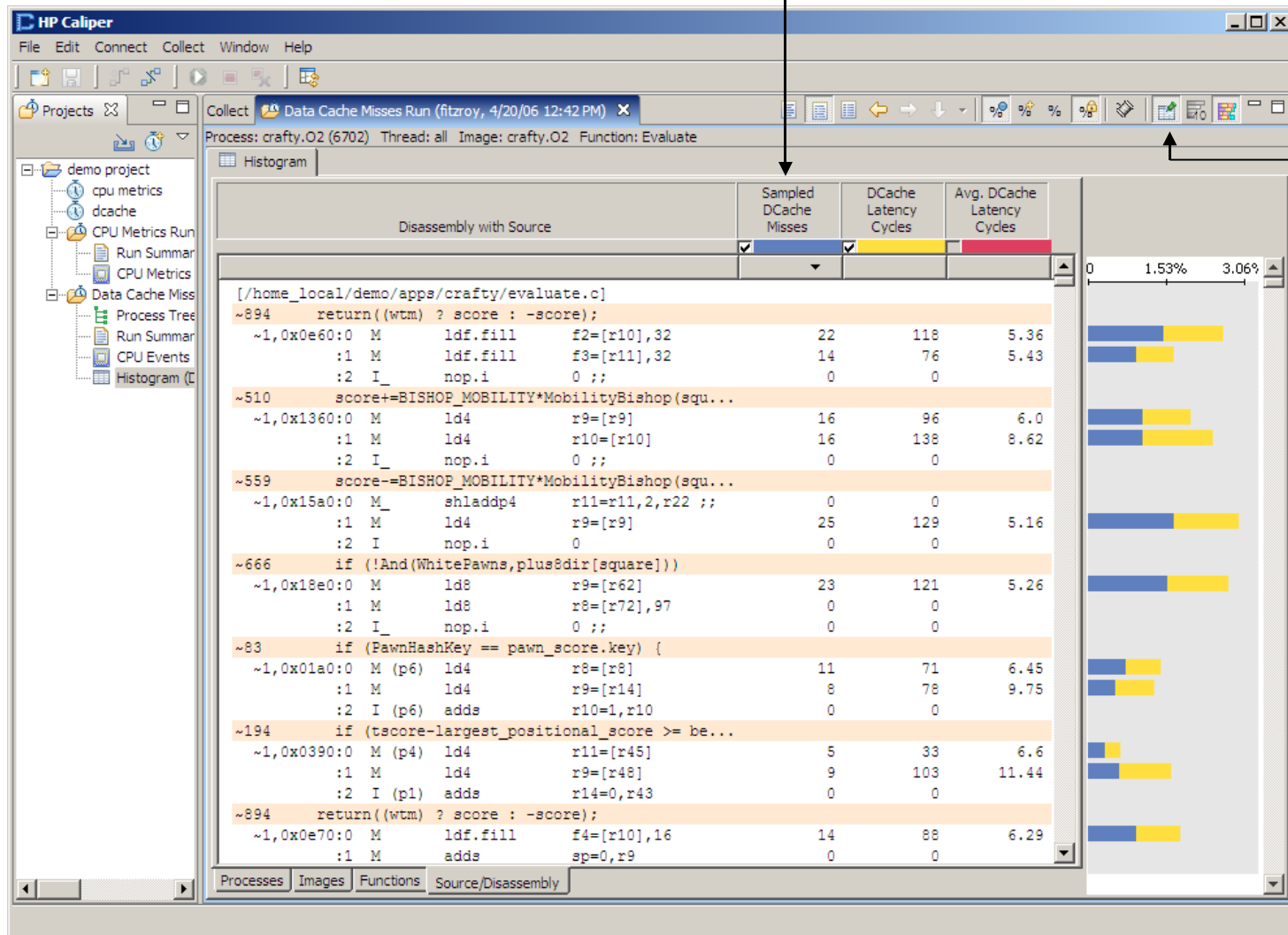
Graph Not Pinned to Show Entire Function

Click Graph to Show Hot Spots in Table

Highlight Shows Table's Current View

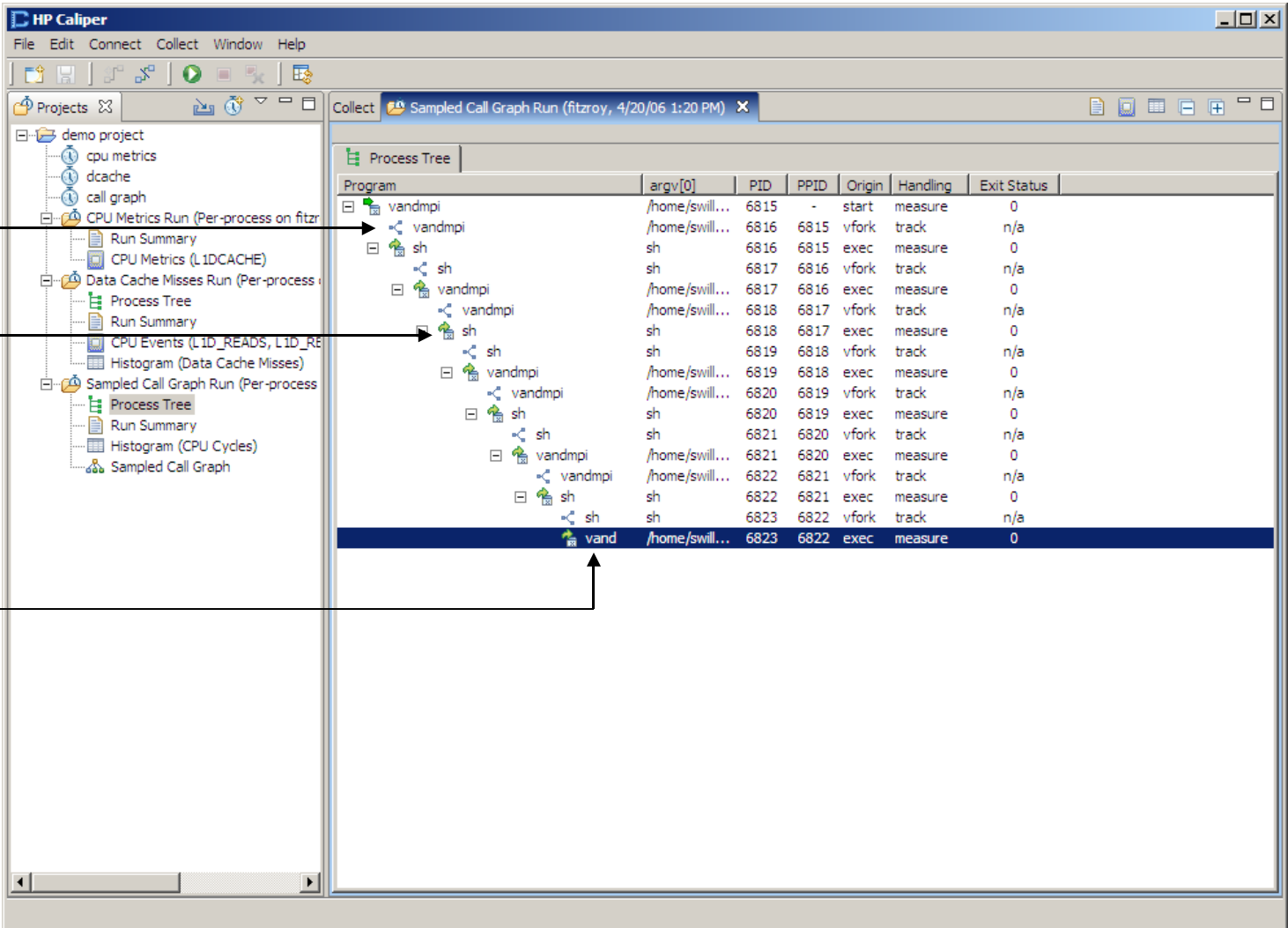
Sort by Cache Misses, Pinned Graph

Instruction Bundles Sorted by Data Cache Misses



Graph
Pinned
to Table

Viewing a Process Tree

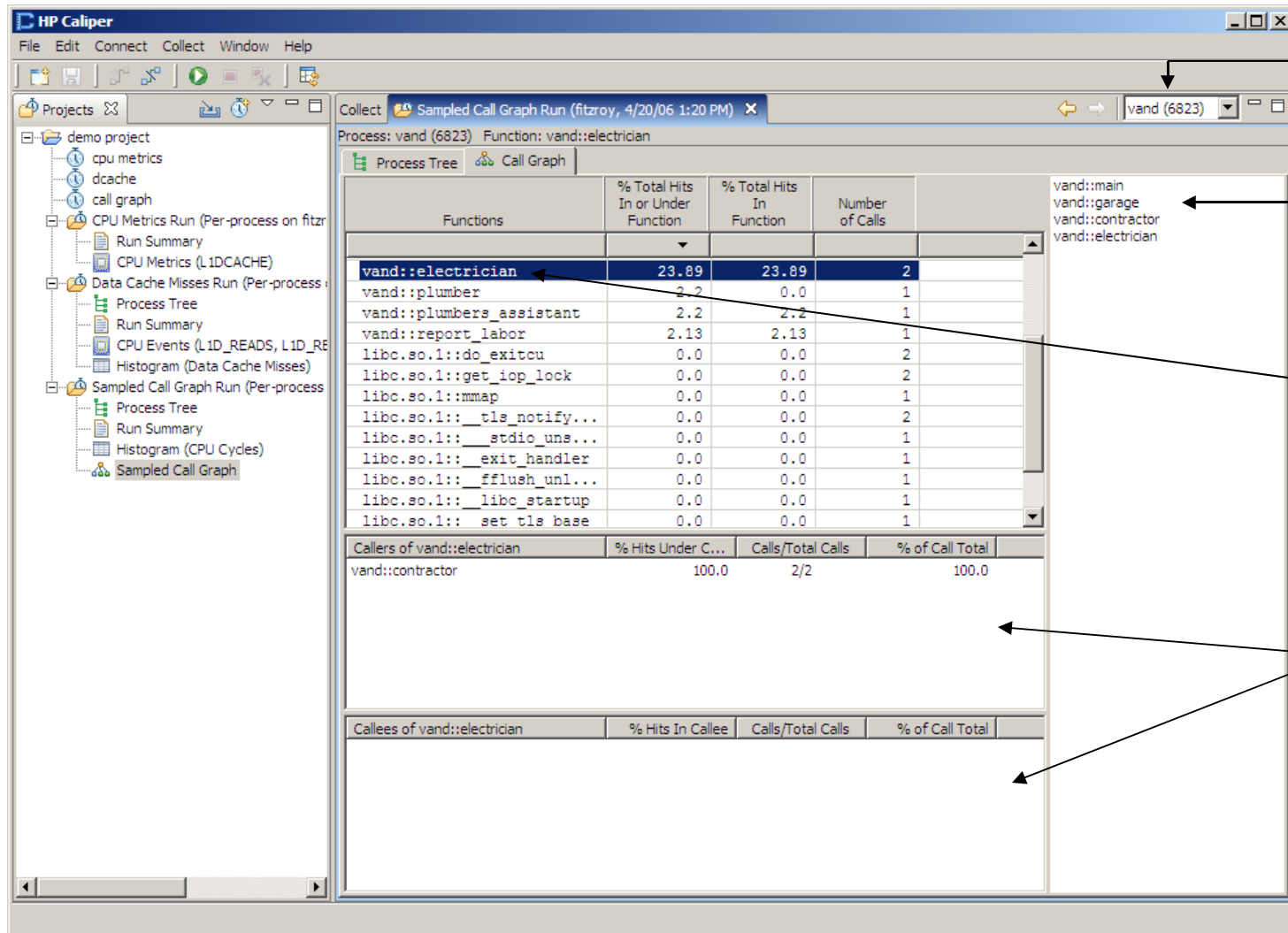


The screenshot shows the HP Caliper interface with a project named 'demo project'. The left pane lists various data collection methods, including 'CPU Metrics Run', 'Data Cache Misses Run', and 'Sampled Call Graph Run'. The right pane displays a 'Process Tree' table with columns for Program, argv[0], PID, PPID, Origin, Handling, and Exit Status. The table lists several processes, including 'vandmpi' and 'sh'. A process named 'vand' is highlighted at the bottom of the table. Annotations on the left side of the image point to specific elements: 'vfork()' and 'exec()' symbols in the left pane, and a 'vand' process in the right pane.

Process Tree Table:

Program	argv[0]	PID	PPID	Origin	Handling	Exit Status
vandmpi	/home/swill...	6815	-	start	measure	0
vandmpi	/home/swill...	6816	6815	vfork	track	n/a
sh	sh	6816	6815	exec	measure	0
sh	sh	6817	6816	vfork	track	n/a
vandmpi	/home/swill...	6817	6816	exec	measure	0
vandmpi	/home/swill...	6818	6817	vfork	track	n/a
sh	sh	6818	6817	exec	measure	0
sh	sh	6819	6818	vfork	track	n/a
vandmpi	/home/swill...	6819	6818	exec	measure	0
vandmpi	/home/swill...	6820	6819	vfork	track	n/a
sh	sh	6820	6819	exec	measure	0
sh	sh	6821	6820	vfork	track	n/a
vandmpi	/home/swill...	6821	6820	exec	measure	0
vandmpi	/home/swill...	6822	6821	vfork	track	n/a
sh	sh	6822	6821	exec	measure	0
sh	sh	6823	6822	vfork	track	n/a
vand	/home/swill...	6823	6822	exec	measure	0

Call Graph Viewer



HP Caliper

File Edit Connect Collect Window Help

Projects demo project

- cpu metrics
- dcache
- call graph
- CPU Metrics Run (Per-process on fitz)
- Run Summary
- CPU Metrics (L1DCACHE)
- Data Cache Misses Run (Per-process)
- Process Tree
- Run Summary
- CPU Events (L1D_READS, L1D_RE)
- Histogram (Data Cache Misses)
- Sampled Call Graph Run (Per-process)
- Process Tree
- Run Summary
- Histogram (CPU Cycles)
- Sampled Call Graph

Collect Sampled Call Graph Run (fitzroy, 4/20/06 1:20 PM)

Process: vand (6823) Function: vand::electrician

Process Tree Call Graph

Functions	% Total Hits In or Under Function	% Total Hits In Function	Number of Calls
vand::electrician	23.89	23.89	2
vand::plumber	2.2	0.0	1
vand::plumbers_assistant	2.2	2.2	1
vand::report_labor	2.13	2.13	1
libc.so.1::do_exitcu	0.0	0.0	2
libc.so.1::get_iop_lock	0.0	0.0	2
libc.so.1::mmap	0.0	0.0	1
libc.so.1::__tls_notify...	0.0	0.0	2
libc.so.1::__stdio_uns...	0.0	0.0	1
libc.so.1::__exit_handler	0.0	0.0	1
libc.so.1::__fflush_unl...	0.0	0.0	1
libc.so.1::__libc_startup	0.0	0.0	1
libc.so.1::__set_tls_base	0.0	0.0	1

Callers of vand::electrician

	% Hits Under C...	Calls/Total Calls	% of Call Total
vand::contractor	100.0	2/2	100.0

Callees of vand::electrician

	% Hits In Callee	Calls/Total Calls	% of Call Total
--	------------------	-------------------	-----------------

Drill-down History

- vand::main
- vand::garage
- vand::contractor
- vand::electrician

Thanks - Those without whom...

- Processor designers
 - Jim Callister of Intel
- Perfmon subsystem maintainers
 - Stephane Eranian (Linux)
 - Dan Truong (HP-UX)

Resources

- caliper-help@cup.hp.com
 - Caliper “help” mailing address
 - Caliper questions, bug reports, enhancement requests
- <http://hp.com/go/caliper>
 - Caliper external website
 - Downloads, documentation, tech notes
- <http://devresource.hp.com>
 - Developers’ Resource website
 - Training webinars
- [Gelato meeting, May 2005 Caliper slides](#)
 - http://www.gelato.org/pdf/may2005/gelato_may2005_caliper_gourio_u_hp.pdf

Questions